

VIRTUAL EXECUTION ENVIRONMENT ACTUALIZATION SYSTEM

Patent Number: JP8063363
Publication date: 1996-03-08
Inventor(s): IMAIZUMI MINORU
Applicant(s):: FUJITSU LTD
Requested Patent: ☐ JP8063363
Application Number: JP19940201074 19940825
Priority Number(s):
IPC Classification: G06F9/46
EC Classification:
Equivalents:

Abstract

PURPOSE: To provide the system, which actualizes virtual execution environment wherein application software is executed without modifying its source program, on an existent operating system.
CONSTITUTION: The source program 11 of the application software 10 has substitute information description parts 12a and 12b, wherein information for replacing procedures, etc., described in a program main body 11 with other procedures by kinds of existent operating systems where virtual execution environment is actualized is described, and are equipped with translation parts 21a and 21b which translate the substitution information described in the program main body and the substitution information description parts 12a and 12b into an executable program that can be executed in the virtual execution environment by utilizing a compiler for the existent operating systems and execution parts 22a and 22b which executes instructions executed by the executable program by utilizing the existent operating systems 60a and 60b.

Data supplied from the esp@cenet database - I2

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-63363

(43) 公開日 平成8年(1996)3月8日

(51) Int.Cl.⁶

G 0 6 F 9/46

識別記号

3 5 0

庁内整理番号

7737-5B

F I

技術表示箇所

審査請求 未請求 請求項の数 5 O L (全 8 頁)

(21) 出願番号 特願平6-201074

(22) 出願日 平成6年(1994)8月25日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中1015番地

(72) 発明者 今泉 稔

群馬県前橋市間屋町1丁目8番地3 株式

会社富士通ターミナルシステムズ内

(74) 代理人 弁理士 遠山 勉 (外1名)

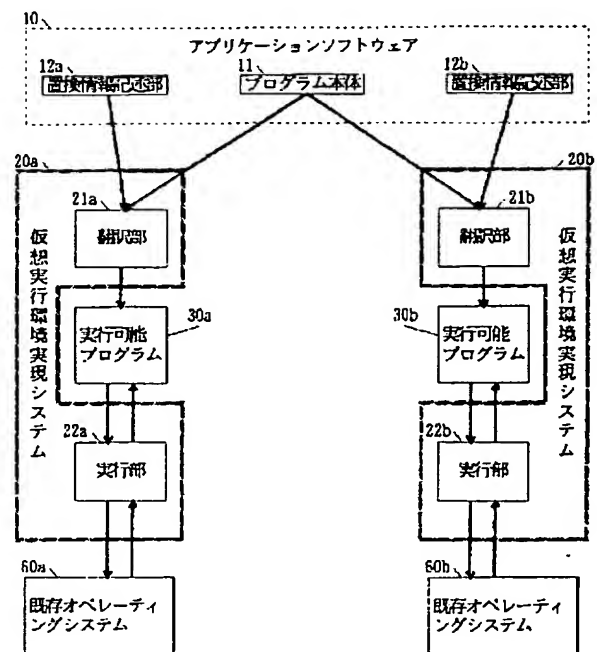
(54) 【発明の名称】 仮想実行環境実現システム

(57) 【要約】

【目的】 アプリケーションソフトウェアをソースプログラムの変更無しに実行可能とする仮想実行環境を既存オペレーティングシステム上に実現するシステムを提供することを目的とする。

【構成】 アプリケーションソフトウェアのソースプログラムは、プログラム本体中に記述された手続等を、仮想実行環境が実現される既存オペレーティングシステムの種別毎に他の手続等に置き換える情報を記述した置換情報記述部を有し、プログラム本体と置換情報記述部に記述された置換情報を、既存オペレーティングシステム用コンパイラを利用して、仮想実行環境上で実行可能な実行可能プログラムに翻訳する翻訳部と、既存オペレーティングシステムを利用して実行可能プログラムが実行する命令を実行する実行部とを備えるように構成した。

本発明の仮想実行環境実現システムの原理図



【特許請求の範囲】

【請求項 1】 プログラミング言語で記述されたプログラム本体を有するソースプログラムから生成されるアプリケーションソフトウェアを、前記プログラム本体の変更無しに実行可能とする仮想実行環境を各種の既存オペレーティングシステム上に実現するシステムにおいて、前記アプリケーションソフトウェアのソースプログラムは、プログラム本体中に記述された手続、関数、論理を、前記仮想実行環境が実現される前記既存オペレーティングシステムの種別毎に他の手続、関数、論理に置き換える情報を記述した置換情報記述部を有し、前記プログラム本体と前記置換情報記述部に記述された置換情報を、既存オペレーティングシステム用コンパイラを利用して、前記仮想実行環境上で実行可能な実行可能プログラムに翻訳する翻訳部と、前記既存オペレーティングシステムを利用して前記実行可能プログラムが実行する命令を実行する実行部とを備えたことを特徴とする仮想実行環境実現システム。

【請求項 2】 請求項 1 において、前記実行部は、前記実行可能プログラムの実行命令が前記既存オペレーティングシステムにあるか否かを判断し、あると判断した場合に、前記既存オペレーティングシステムを利用して実行することを特徴とする仮想実行環境実現システム。

【請求項 3】 請求項 2 において、前記実行部は、前記実行可能プログラムの実行命令が前記既存オペレーティングシステムにないと判断した場合に、前記既存オペレーティングシステムの機能を組み合わせて実現できるか否かを判断し、実現できると判断した場合に、前記既存オペレーティングシステムの機能を組み合わせて実行することを特徴とする仮想実行環境実現システム。

【請求項 4】 請求項 3 において、前記実行部は、前記実行可能プログラムの実行命令が、前記既存オペレーティングシステムの機能を組み合わせて実現できないと判断した場合に、前記既存オペレーティングシステムの機能を利用せずに、実行することを特徴とする仮想実行環境実現システム。

【請求項 5】 請求項 1 から 4 において、前記翻訳部は、前記既存オペレーティングシステムが直接実行可能な命令に対しては、前記既存オペレーティングシステムに直接アクセスして実行する実行可能プログラムに翻訳することを特徴とする仮想実行環境実現システム。

【発明の詳細な説明】**【0001】**

【産業上の利用分野】 本発明は、アプリケーションソフトウェアを、ソースプログラムの変更無しに実行可能とする仮想実行環境を各種の既存オペレーティングシステム上に実現するシステムに関する。

【0002】

【従来の技術】 アプリケーションソフトウェアは、動作速度の点からは、動作する個々の計算機システム毎に書き換えられて使用されることが望ましい。

【0003】 しかし、個々の計算機システムの数だけアプリケーションソフトウェアを用意するのは、流通性、経済性等の観点から好ましくない。そこで、従来から、下記のような方法がとられてきている。

【0004】 第一は、計算機システムに構築されたオペレーティングシステム（以下、OS という）毎にアプリケーションソフトウェアを対応させる方法である。この場合、図 7 に示すような構成となる。

【0005】 第二は、ある CPU（A）用の OS（B）に対応して書かれたアプリケーションを、下記の方法で、異種の CPU（C）用の OS（D）上でも動作させる方法である。即ち、OS（D）上に、CPU（A）の命令をシミュレートするシミュレート部を設けるとともに、このシミュレート部上で OS（B）を実行し、この OS（B）上でアプリケーションを実行する方法である。この場合、図 8 に示すような構成となる。

【0006】

【発明が解決しようとする課題】 第一の方法では、OS 毎の差異をアプリケーションソフトウェア側で吸収する必要があるため、アプリケーションのソースプログラムを OS 毎に書き換えねばならず、書換時にバグが入ったり、全ての OS 向けに同時にリリースすることが難しいなどの問題がある。

【0007】 また、第二の方法では、第一の方法の問題は避けられるが、シミュレート部を介して実行するため、動作速度が遅くなりやすい問題がある。本発明は、このような事情に鑑みてなされたもので、アプリケーションソフトウェアをソースプログラムの変更無しに実行可能とする仮想実行環境を各種の既存オペレーティングシステム上に実現するシステムを提供することを第一の課題とする。

【0008】 また、本発明は、アプリケーションソフトウェアをソースプログラムの変更無しに実行可能とする共に、動作速度を犠牲にすることのない仮想実行環境を各種の既存オペレーティングシステム上に実現するシステムを提供することを第二の課題とする。

【0009】**【課題を解決するための手段】**

<本発明の第 1 の仮想実行環境実現システム> 本発明の第 1 の仮想実行環境実現システムは、前述した第一の課題を解決するため、図 1 に示す原理図のように下記の如く構成されている。

【0010】 即ち、プログラミング言語で記述されたプログラム本体（11）を有するソースプログラムから生成されるアプリケーションソフトウェア（10）を、前記プログラム本体（11）の変更無しに実行可能とする

仮想実行環境を各種の既存オペレーティングシステム（60）上に実現するシステムにおいて、前記アプリケーションソフトウェア（10）のソースプログラムは、プログラム本体中（11）に記述された手続、関数、論理を、前記仮想実行環境が実現される前記既存オペレーティングシステム（60）の種別毎に他の手続、関数、論理に置き換える情報を記述した置換情報記述部（12）を有し、前記プログラム本体（11）と前記置換情報記述部（12）に記述された置換情報を、既存オペレーティングシステム用コンパイラを利用して、前記仮想実行環境上で実行可能な実行可能プログラム（30）に翻訳する翻訳部（21）と、前記既存オペレーティングシステムを利用して前記実行可能プログラム（30）が実行する命令を実行する実行部（22）とを備えたことを特徴とする（請求項1に対応）。

【0011】ここで、既存オペレーティングシステム（60）は、Windows-NTやUNIX、SOLARIS、MS-DOS等を利用することができる。また、プログラム本体（11）は、C、FORTRAN、BASIC等の汎用プログラミング言語の他に、ユーザが設計した言語等を利用することができる。

【0012】そして、実行可能プログラム（30）は、同種類の既存オペレーティングシステム（60）に構築された仮想実行環境実現システム（20）上でのみ実行可能となる。即ち、アプリケーションソフトウェア（10）は、ソースレベルでのみ互換性が保たれ、実行レベルでは互換性が失われる。

【0013】なお、図1は、第一の既存オペレーティングシステム（60a）及び第二の既存オペレーティングシステム（60b）上に仮想実行環境を実現した例を示しているが、一つの既存オペレーティングシステム上にだけ実現しても良いし、3つ以上の既存オペレーティングシステム上に実現しても良い。

【0014】また、プログラム本体（11）がC言語で記述される場合、置換情報記述部（12）に記述されるデータは、例えば、マクロ定義を利用することができる。

<本発明の第2の仮想実行環境実現システム>本発明の第2の仮想実行環境実現システムは、前述した第一の課題を解決するため、下記の如く構成されている。

【0015】即ち、第1の仮想実行環境実現システムにおいて、前記実行部（22）は、前記実行可能プログラム（30）の実行命令が前記既存オペレーティングシステム（60）にあるか否かを判断し、あると判断した場合に、前記既存オペレーティングシステム（60）を利用して実行することを特徴とする（請求項2に対応）。

【0016】<本発明の第3の仮想実行環境実現システム>本発明の第3の仮想実行環境実現システムは、前述した第一の課題を解決するため、下記の如く構成されている。

【0017】即ち、第2の仮想実行環境実現システムにおいて、前記実行部（22）は、前記実行可能プログラム（30）の実行命令が前記既存オペレーティングシステム（60）にないと判断した場合に、前記既存オペレーティングシステム（60）の機能を組み合わせて実現できるか否かを判断し、実現できると判断した場合に、前記既存オペレーティングシステム（60）の機能を組み合わせて実行することを特徴とする（請求項3に対応）。

【0018】<本発明の第4の仮想実行環境実現システム>本発明の第4の仮想実行環境実現システムは、前述した第一の課題を解決するため、下記の如く構成されている。

【0019】即ち、第3の仮想実行環境実現システムにおいて、前記実行部（22）は、前記実行可能プログラム（30）の実行命令が、前記既存オペレーティングシステム（60）の機能を組み合わせて実現できないと判断した場合に、前記既存オペレーティングシステム（60）の機能を利用せずに、実行することを特徴とする（請求項4に対応）。

【0020】<本発明の第5の仮想実行環境実現システム>本発明の第5の仮想実行環境実現システムは、前述した第二の課題を解決するため、下記の如く構成されている。

【0021】即ち、第1から第4の仮想実行環境実現システムにおいて、前記翻訳部（21）は、前記既存オペレーティングシステム（60）が直接実行可能な命令に対しては、前記既存オペレーティングシステム（60）に直接アクセスして実行する実行可能プログラム（30）に翻訳することを特徴とする（請求項5に対応）。

【0022】

【作用】

<本発明の第1の仮想実行環境実現システムの作用>本発明の第1の仮想実行環境実現システムによれば、翻訳部（21）は、プログラム本体（11）と置換情報記述部（12）に記述された置換情報を、既存オペレーティングシステム用コンパイラを利用して、仮想実行環境上で実行可能な実行可能プログラム（30）に翻訳する。

【0023】そして、実行部（22）は、既存オペレーティングシステム（60）を利用して実行可能プログラム（30）を実行する。

<本発明の第2の仮想実行環境実現システムの作用>本発明の第2の仮想実行環境実現システムによれば、第1の仮想実行環境実現システムの作用に加えて、実行部（22）は、実行可能プログラム（30）の実行命令が既存オペレーティングシステム（60）にあるか否かを判断し、あると判断した場合に、既存オペレーティングシステム（60）を利用して実行する。

【0024】<本発明の第3の仮想実行環境実現システムの作用>本発明の第3の仮想実行環境実現システムに

よれば、第2の仮想実行環境実現システムの作用に加えて、実行部(22)は、実行可能プログラム(30)の実行命令が既存オペレーティングシステム(60)にない

と判断した場合に、既存オペレーティングシステム(60)の機能を組み合わせて実現できるか否かを判断し、実現できると判断した場合に、既存オペレーティングシステム(60)の機能を組み合わせて実行する。

【0025】<本発明の第4の仮想実行環境実現システムの作用>本発明の第4の仮想実行環境実現システムによれば、第3の仮想実行環境実現システムの作用に加えて、実行部(22)は、実行可能プログラム(30)の実行命令が、既存オペレーティングシステム(60)の機能を組み合わせて実現できないと判断した場合に、既存オペレーティングシステム(60)の機能を利用せずに、実行する。

【0026】<本発明の第5の仮想実行環境実現システムの作用>本発明の第5の仮想実行環境実現システムによれば、第1から第4の仮想実行環境実現システムの作用に加えて、翻訳部(21)は、既存オペレーティングシステム(60)が直接実行可能な命令に対しては、既存オペレーティングシステム(60)に直接アクセスして実行する実行可能プログラムに翻訳する。

【0027】

【実施例】以下、本発明の2つの実施例を図面を参照して説明する。

【0028】

【第1実施例】第1実施例は、第一の計算機システム(A)に構築された既存OS-A(60a)上と、第二の計算機システム(B)に構築された既存OS-B(60b)上に、仮想実行環境を実現する例である。

【0029】<第1実施例の構成>図2には、第1実施例のシステム構成図が示されている。以下、システムの各構成要素について説明する。

【0030】(計算機システムA) 計算機システムAは、既存CPU-A(50a)を有する。この既存CPU-A(50a)上で、既存OS-A(60a)が動作する。この既存OS-A(60a)上に、実行部としての仮想実行環境実現部(22a)が設けられている。この仮想実行環境実現部(22a)は、既存OS-A(60a)から見れば一種のアプリケーションプログラム(応用プログラム)であるが、計算機システムAを利用する側から見れば、既存OS-A(60a)に代わる別の実行環境、言い換えれば別のOS環境の一部を提供している。そして、仮想実行環境実現部(22a)上では、n個の仮想実行環境用アプリケーション(30a-1~30a-n)が動作可能な状態となっている。これらの、仮想実行環境用アプリケーション(30a-1~30a-n)は、シングルタスクで動作させることもできるし、マルチタスクで動作させることもできる。

【0031】仮想実行環境用アプリケーション(30a

-1~30a-n)は、プログラム本体(11-1~11-n)と既存OS-A(60a)用の仮想実行環境インタフェース定義データ(12a)を、既存OS-A用コンパイラ(21a)によりコンパイルすることにより生成される。つまり、既存OS-A用コンパイラ(21a)は、翻訳システムの役割をしている。

【0032】(計算機システムB) 計算機システムBは、既存CPU-B(50b)を有する。この既存CPU-B(50b)上で、既存OS-B(60b)が動作する。この既存OS-B(60b)上に、実行部としての仮想実行環境実現部(22b)が設けられている。この仮想実行環境実現部(22b)は、既存OS-B(60b)から見れば一種のアプリケーションプログラム(応用プログラム)であるが、計算機システムBを利用する側から見れば、既存OS-B(60b)に代わる別の実行環境、言い換えれば別のOS環境の一部を提供している。そして、仮想実行環境実現部(22b)上では、n個の仮想実行環境用アプリケーション(30b-1~30b-n)が動作可能な状態となっている。これらの、仮想実行環境用アプリケーション(30b-1~30b-n)は、シングルタスクで動作させることもできるし、マルチタスクで動作させることもできる。

【0033】仮想実行環境用アプリケーション(30b-1~30b-n)は、プログラム本体(11-1~11-n)と既存OS-B(60b)用の仮想実行環境インタフェース定義データ(12b)を、既存OS-B用コンパイラ(21b)によりコンパイルすることで生成される。つまり、既存OS-B用コンパイラ(21b)は、翻訳システムの役割をしている。

【0034】<第1実施例の動作>第1実施例の動作として、計算機システムAにおける仮想実行環境実現部(22a)の処理を図3を参照して説明する。図3は、仮想実行環境実現部の処理フローチャートである。なお以下の説明は、計算機システムBにそのままあてはめることができる。

【0035】まず、仮想実行環境用アプリケーション(30a-1~30a-n)が命令を実行する(仮想実行環境実現部(22a)に対して命令を呼び出す)と、実行する命令と同等の機能が既存OS-A(60a)に存在するか否かを判断する(ステップ301)。

【0036】ステップ301で「YES」(存在する)と判断された場合に、既存OS-A(60a)の同等機能を利用して実行する(ステップ307)。そして、既存OS-A(60a)の復帰情報を基に仮想実行環境実現部(22a)への復帰情報を設定する(ステップ308)。この後、命令を実行した仮想実行環境用アプリケーション(30a-1~30a-n)に復帰する(ステップ309)。

【0037】ステップ301で「NO」(存在しない)と判断された場合に、既存OS-A(60a)の機能を

組み合わせて実現可能な否かを判断する（ステップ302）。

【0038】ステップ302で「YES」（実現できる）と判断された場合に、既存OS-A（60a）の機能を組み合わせて実行する（ステップ303）。そして、既存OS-A（60a）の復帰情報を基に仮想実行環境実現部（22a）への復帰情報を設定する（ステップ304）。この後、命令を実行した仮想実行環境用アプリケーション（30a-1～30a-n）に復帰する（ステップ309）。

【0039】ステップ302で「NO」（実現できない）と判断された場合に、仮想実行環境実現部（22a）内で実行する（ステップ305）。そして、結果を復帰情報に設定する（ステップ306）。この後、命令を実行した仮想実行環境用アプリケーション（30a-1～30a-n）に復帰する（ステップ309）。

【0040】＜仮想実行環境インタフェース定義データの役割の概念＞次に、仮想実行環境インタフェース定義データの役割の概念について、図5を参照して説明する。

【0041】まず、仮想実行環境用アプリケーションソースプログラム（30）には、処理Aを呼び出す命令があり、仮想実行環境インタフェース定義データ（21）には、処理Aを「abc呼び出し」に変換する情報があるものとする。

【0042】既存OS用コンパイラ（21）は、プログラム本体（10）と仮想実行環境インタフェース定義データ（11）を基に、仮想実行環境用アプリケーション（11）を生成する。

【0043】仮想実行環境用アプリケーション（30）は、プログラム本体（11）で「処理A呼び出し」と記述されていた命令を、実際には、仮想実行環境実現部（22）に対する「abc呼び出し」命令として実行する。

【0044】仮想実行環境実現部（22）は、仮想実行環境用アプリケーション（30）から「abc呼び出し」を受けると、既存OS（60）が有する命令であるシステムコール1及び2を発行して、「abc呼び出し」を実行する。

【0045】ここで、「処理A」及び「abc呼び出し」の具体例を説明する。「処理A」は、システムの標準入力装置から1文字を入力する手続きであるとする。通常、標準入力装置はキーボードであることが多いが、音声マイク、通信装置、CCDカメラ等を標準入力とするシステムも考えられる。そこで、プログラム本体（11）では、単に、システムの標準入力装置から1文字を入力する命令とだけ記述しておき、「abc呼び出し」では、システムの特性に合わせて、音声マイクに入力された音声情報から1文字を抽出する手続きに置換したり、通信装置から得られた通信情報から1文字を読み出

す手続きに置換したりする。

【0046】＜アプリケーション管理方法の概念図＞次に、アプリケーション管理方法の概念について、図6を参照して説明する。まず、仮想実行環境用アプリケーション（301）は、プログラム起動されると、仮想実行環境実現部（22）を呼び出す。呼び出された仮想実行環境実現部（22）は、プログラム管理テーブルを獲得する。このプログラム管理テーブルには、プログラム管理情報や、既存OS（60）から得たプロセス情報が格納されている。

【0047】そして、仮想実行環境実現部（22）は、プログラム管理テーブルから得た情報を基に、既存OS（60）にプロセス起動を依頼する。この依頼の結果、既存OS（60）より得られた情報をプロセス管理テーブルに設定しておく。

【0048】そして、仮想実行環境用アプリケーション（11）に、プログラムIDを通知する。

＜第1実施例の効果＞第1実施例によれば、仮想実行環境インタフェース定義データ（12）や仮想実行環境実現部（22）を、実装されるハードウェアや既存OS（60）の機能に応じて変更することで、プログラム本体（11）の変更無しに実行可能とする仮想実行環境を提供することができる。

【0049】

【第2実施例】第2実施例は、仮想実行環境用アプリケーション（30）が、既存OS（60）の命令（システムコール等）をも直接利用可能なようにする例である。

【0050】＜第2実施例の構成＞図4には、第2実施例のシステム構成図が示されている。このシステムは、既存CPU（50）を有する。この既存CPU（50）上で、既存OS（60）が動作する。この既存OS（60）上に、既存OS（60）から見て一種のアプリケーションプログラムである仮想実行環境実現部（22）が動作している。また、既存OS（60）上には、既存OS用アプリケーション（61-1～61-n）も実行可能な状態にされている。さらに、既存OS（60）上には、仮想実行環境用アプリケーション（30）が実行可能な状態にされている。

【0051】＜第2実施例の動作・効果＞第2実施例において、仮想実行環境用アプリケーション（30）が、仮想実行環境実現部（22）に命令実行を依頼する場合には、その動作は、第1実施例と同様にして行われる。一方、既存OS（60）に命令実行を依頼する場合には、既存OS（60）用のシステムコールを発行する。

【0052】この第2実施例は、既存OS（60）のシステムコールを利用するため、下記の効果を奏することができる。

（イ）既存OS用アプリケーション（61-1～61-n）と直接連携が行える。このため、既存OS用アプリケーション（61-1～61-n）と連携しながら処理

する分野（例えば、通信等）において有効である。

（ロ）仮想実行環境実現部（22）がサポートしていない、既存OS（60）に特有の機能やハードウェアを活用することができる。

【0053】

【発明の効果】本発明の第一の仮想実行環境実現システムによれば、既存オペレーティングシステムの種別毎に、置換情報記述部の記述情報と実行部を用意することで、アプリケーションソフトウェアのプログラム本体を変更せずに実行可能とする仮想実行環境を各種の既存オペレーティングシステム上に実現することができる。

【0054】また、本発明の第二の仮想実行環境実現システムによれば、仮想実行環境上で実行されるアプリケーションソフトウェアが、既存OSに特有の機能やハードウェアを活用することで、アプリケーションソフトウェアをソースプログラムの変更無しに実行可能とする共に、動作速度を犠牲にすることのない仮想実行環境を各種の既存オペレーティングシステム上に実現することができる。

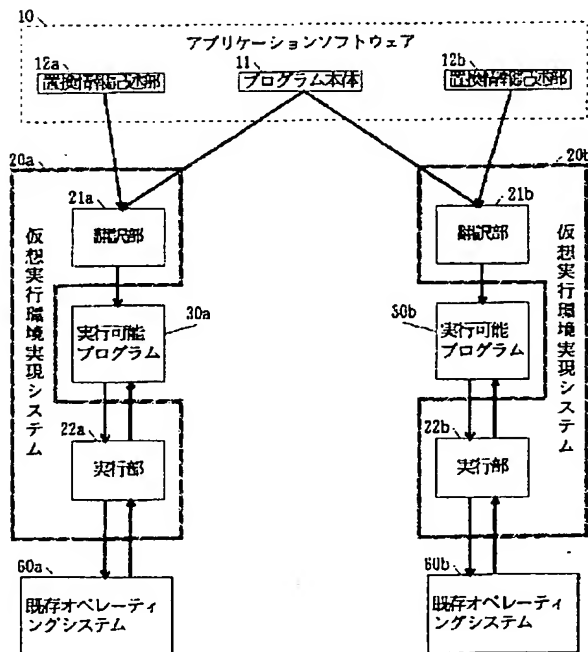
【図面の簡単な説明】

【図1】本発明の仮想実行環境実現システムの原理図である。

【図2】第1実施例のシステム構成図である。

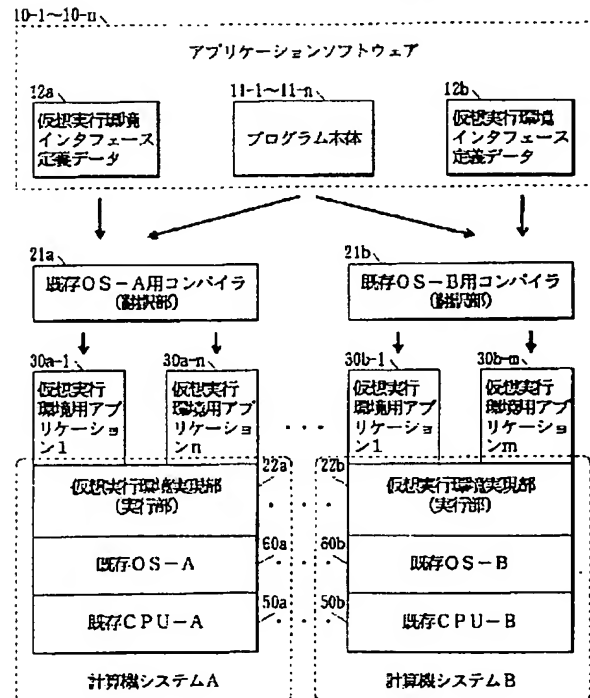
【図1】

本発明の仮想実行環境実現システムの原理図



【図2】

第1実施例のシステム構成図



【図3】第1実施例における仮想実行環境実現部の処理フローチャートである。

【図4】第2実施例のシステム構成図である。

【図5】仮想実行環境インタフェース定義データ（置換情報記述部に記述されたデータ）の役割の概念図である。

【図6】アプリケーション管理方法の概念図である。

【図7】従来の一般的な計算機システムの構成図である。

【図8】異種CPU用のアプリケーションを動作可能とする従来の計算機システムの構成図である。

【符号の説明】

10・・・アプリケーションソフトウェア

11・・・プログラム本体

12・・・置換情報記述部

20・・・仮想実行環境実現システム

21・・・翻訳部（既存OS用コンパイラ）

22・・・実行部（仮想実行環境実現部）

30・・・仮想実行環境用アプリケーション

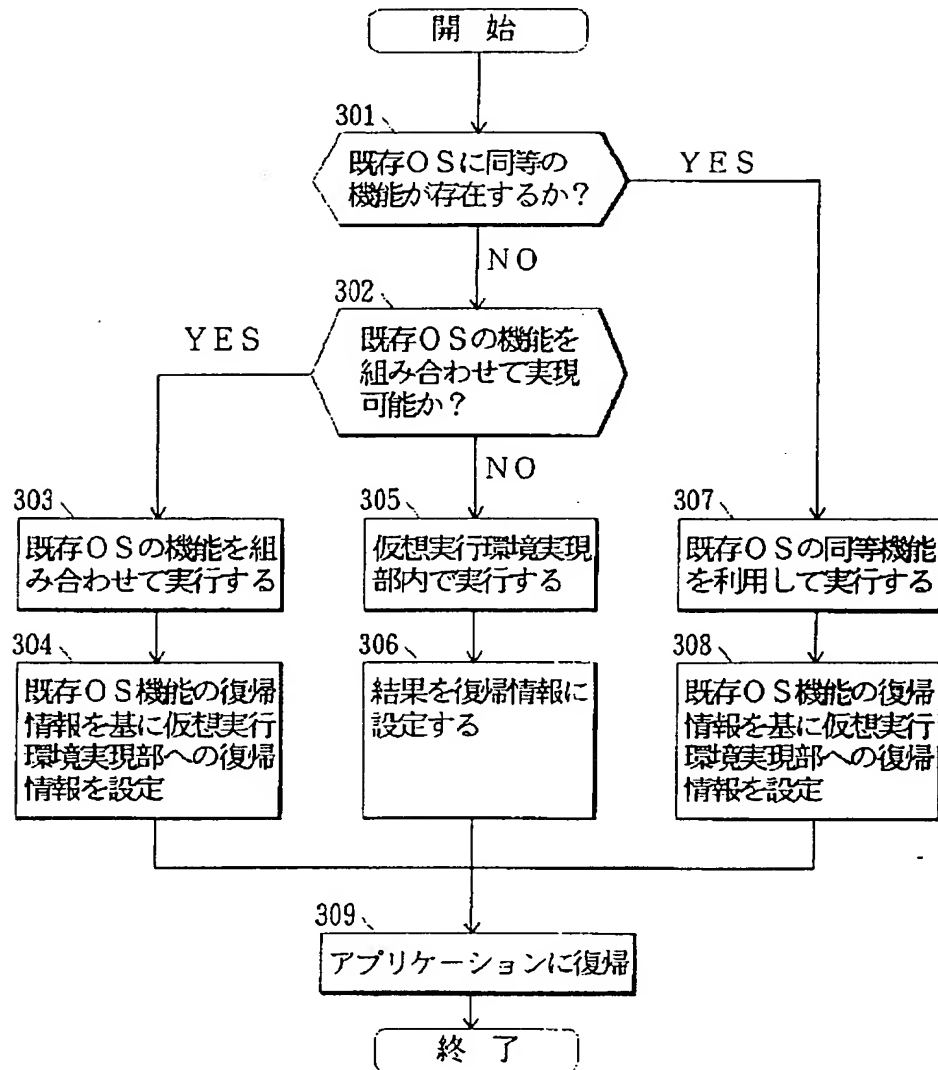
50・・・既存CPU

60・・・既存OS

61-1～61-n・・・既存OS用アプリケーション

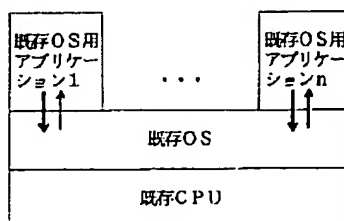
【図 3】

第 1 実施例における仮想実行環境実現部の 処理フローチャート



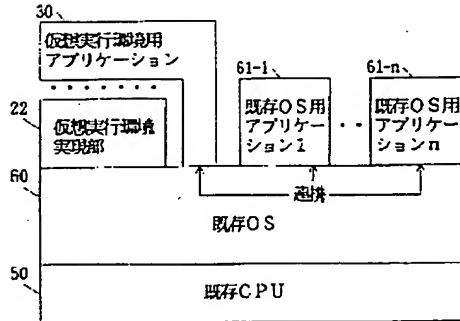
【図 7】

従来一般的な計算機システムの構成図



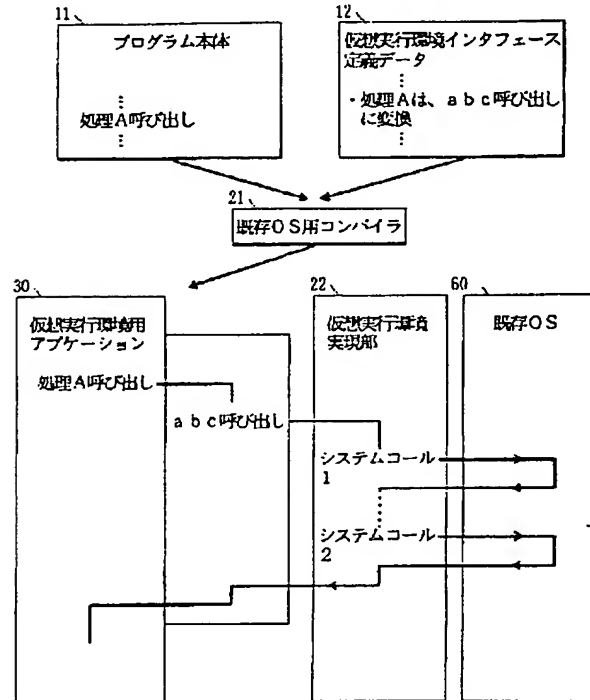
【図 4】

第 2 実施例のシステム構成図



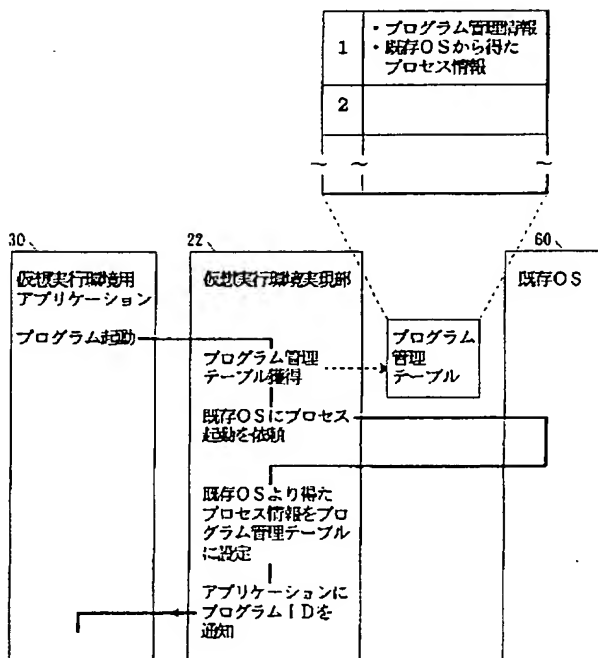
【図 5】

仮想実行環境インタフェース定義データ（置換情報記述部に記述されたデータ）の役割の概念図



【図 6】

アプリケーション管理方法の概念図



【図 7】

異種 CPU 用のアプリケーションを動作可能とする従来の計算機システムの構成図

